REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-01-

0246

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPO. Final Technical Report 01 Apr 96 - 31 Mar 00

4. TITLE AND SUBTITLE
Compositional Analysis of Expected Delay in Networks of Automata

5. FUNDING NUMBERS
F49620-96-1-0087

6. AUTHOR(S)
Scott A. Smolka

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Dept of Computer Science
State University of New York
Stony Brook, NY 11794

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
801 N. Randolph St, Rm 732
Arlington, VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
F49620-96-1-0087

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

13. ABSTRACT *(Maximum 200 words)*

We have succeeded in extending our earlier work on probabilistic I/O automata to obtain a practical method for computing termination probabilities and exptected termination times in networks of PIOA. Our method is compositional, avoiding the construction of the network's global state space, and as a result very efficient. This represents the culmination of work we first reported in last year's annual report.

20010426 049

14. SUBJECT TERMS	15. NUMBER OF PAGES 7
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Compositional Analysis of Expected Delay in Networks of Automata
F49620-96-1-0087

Final Report

Scott A. Smolka and **Eugene W. Stark**
Department of Computer Science
SUNY at Stony Brook
Stony Brook, NY 11794-4400
(516) 632-8453 (voice)
(516) 632-8334 (fax)
sas@cs.sunysb.edu, stark@sunysb.edu

Stephanie White
Northrop Grumman
MS K04-14
Bethpage, NY 11714
(516) 575-2201 (voice)
(516) 575-4864 (fax)
whitest@mail.northgrum.com

OBJECTIVES

Probabilistic I/O Automata (PIOA), a natural extension of the I/O Automaton model of Lynch and Tuttle, have been developed by the PIs to model and analyze distributed systems that exhibit behavior that is statistical or probabilistic in nature and subject to timing constraints. The objectives of this grant are to:

1. continue our investigation of the probabilistic I/O automaton model with an emphasis on developing techniques that will permit the probabilistic behavior and delay in distributed systems to be analyzed compositionally;
2. implement the resulting techniques in the Concurrency Factory, a CASE environment for the design, specification, verification, and implementation of asynchronous distributed systems; and
3. perform a technology transfer with Northrop Grumman by applying our techniques to several applications currently of interest to Northrop Grumman.

STATUS OF EFFORT

Compositional Analysis of PIOAs

We have succeeded in extending our earlier work on probabilistic I/O automata to obtain a practical method for computing termination probabilities and expected termination times in networks of PIOA. Our method is compositional, avoiding the construction of the network's global state space, and as a result very efficient. This represents the culmination of work we

first reported in last year's annual progress report.

Implementation and Initial Benchmarking of Compositional Methods

We have implemented a suite of routines for constructing and manipulating PIOA's, for use as a testbed for our analysis techniques.

Technology Transfer with Northrop Grumman

We have used a computer-aided verification tool to model and analyze the Display LAN protocol of the Northrop Grumman E-2C surveillance aircraft. A race condition was identified. Although of interest to the protocol designers, it was subsequently deemed not to impact the practical operation of the protocol.

ACCOMPLISHMENTS/NEW FINDINGS

We briefly discuss our accomplishments in the areas of probabilistic I/O automata and protocol verification.

Compositional Analysis for PIOAs

We have devised a practical, compositional algorithmic technique for determining *completion probability* and *expected completion time* properties of systems of PIOAs. This represents the culmination of work we first reported in last year's annual progress report. The results of our research have been published in a paper that appeared in the *Proceedings of the 1998 IEEE Symposium on Logic in Computer Science* (LICS '98). In what follows, we first describe briefly the main features of the PIOA model. This is followed by an overview of our compositional methods.

Probabilistic I/O automata (PIOA) are an adaptation of the I/O automata model developed by Nancy Lynch and her students, which they have successfully applied to the hierarchical specification and verification of distributed algorithms. An I/O automaton is basically a transition system, consisting of a set of states, a set of actions, and a transition relation, equipped with a partitioning of the actions into *input*, *output*, and *internal* actions. Input actions represent stimuli applied by the environment to the automaton, output actions represent responses by the automaton to the environment, and internal actions represent internal activity of the automaton in which interaction with the environment does not occur. It is required of an I/O automaton that it be *input-enabled*: a transition is possible for every input action from every state of the automaton. This captures the idea that an I/O automaton cannot constrain the ability of its environment to apply stimuli. On the other hand, the occurrence of output or internal actions are completely under the control of the automaton, and these actions are therefore referred to as *locally controlled*.

A key feature of the I/O automaton model is the operation of *composition*, by which a so-called "compatible" collection of I/O automata can be combined into a single, larger automaton. A collection of I/O automata is compatible when their sets of internal actions are pairwise disjoint, and there is no action that is an output action for more than one of the automata in the collection. Under this condition, one can form a composite I/O automaton, whose state set is the cartesian product of the state sets of the

original automata, whose action set is the union of the action sets of the original automata, and whose transition relation reflects the conception of a collection of automata executing concurrently and asynchronously, with interaction between the automata occurring when an output action of one automaton happens to also be an input action of other automata, resulting in a joint transition of all the automata that share that action. The composition operation of I/O automata is what makes them useful for giving hierarchical specifications and verifications of complex systems.

The PIOA model, described by us in a 1997 article in the journal *Theoretical Computer Science*, integrates probability into the I/O automaton model, while carrying over in a natural way the essential features of asynchrony and compositionality. To the original I/O automaton model, two kinds of probability-related information are added. First, probability distributions are associated with each state q of an automaton, in such a way that for each input action a, there is one probability distribution covering all transitions for action a from state q, and such that there is another single probability distribution covering all transitions for locally controlled actions from state q. The probability distribution associated with a particular input action a governs the choice of a particular transition, once it has been determined that the environment will apply an a action as a stimulus in the next step. The probability distribution associated with the locally controlled actions governs the choice of which transition will be taken, once it has been determined that the automaton itself, rather than the environment, will control the next step.

The second type of probabilistic information included in the PIOA model consists of a so-called *delay parameter* associated with each state. The delay parameter is a nonnegative real number, which we interpret as the parameter of an exponentially distributed random variable that describes the amount of time spent by an automaton in a state before it executes the next locally controlled action. In other words, the delay parameter associated with a state describes the rate of departure from that state via a locally controlled action. Delay parameters enable us to "probabilize" the scheduling of locally controlled transitions taken by a system of component PIOAs, according to the following *race criterion*: upon entering a state, each component PIOA chooses randomly, according to the exponential distribution whose parameter is the delay parameter of that state, a nonnegative real number that represents the amount of time that PIOA will remain in this state before executing the next locally controlled action. The times chosen by each of the component PIOAs are compared, the PIOA that has chosen the smallest time is declared "the winner," and it is allowed to perform the next locally controlled action at the time it has chosen.

The above intuition is reflected in the definition of composition for a compatible collection of PIOAs. As for ordinary I/O automata, the state set for the composition of PIOAs is the cartesian product of the state sets of the components. The transition relation for the composition is derived from those of the components in the same way as for ordinary I/O automata. Since the minimum of a collection of exponentially distributed random variables is also exponentially distributed, with a parameter that is the sum of the parameters of the random variables in the collection, the delay parameter associated with a state of the composite automaton is the sum of the delay parameters of each of the constituent states. Moreover, the probability that any particular component automaton will be the next one to execute a locally controlled action from a given global state is given by the ratio of the delay parameter of that automaton to the sum of the delay parameters for all the automata. This leads to a weighting rule for combining the individual probability distributions on the locally controlled actions of the component automata into a single probability distribution on the locally controlled actions of the composite automaton.

We now turn to the problem of analyzing a PIOA, formed as the composition of a compatible collection

of component PIOAs, for certain performance parameters, including: (1) the probability that a PIOA will eventually complete a finite action sequence drawn from a specified (possibly infinite) set of such sequences, and (2) assuming that a PIOA will eventually complete such a sequence with probability one, the expected time it will take until this event occurs. We call (1) the *completion probability* for the specified set of action sequences, and we call (2) the *expected completion time*.

A PIOA is called *closed* if it has an empty set of input actions, and therefore represents an autonomous system that is unable to receive stimuli from its environment. Closed PIOAs are an example of *continuous-time Markov processes*, which have been well studied in the literature, and for which analysis techniques are available for the types of parameters we are interested in. However, a problem with existing Markovian analysis techniques is that they are not compositional: they require as a prerequisite the construction of a global system description such as matrices of transition probabilities and holding-time distributions. Thus, if we wish to use these existing techniques to analyze a closed PIOA described as the composition of a compatible collection of component PIOAs, we must begin by explicitly calculating the composition of the component PIOAs, in order to obtain a global system description suitable for further analysis. As already discussed above, the space required for such a global system description increases exponentially with the number of components, and thus it will be impractical to handle very large systems using this approach.

We have devised a practical, compositional algorithmic technique for determining completion probability and expected completion time properties of systems of PIOAs. An example of a completion probability property in a communications protocol context would be: "the probability that a message issued by the sender is received by the receiver after at most three attempts is 0.5." An example of an expected completion time property would be: "the expected time for a message that has just been issued by the sender to be acknowledged by the receiver is 3 milliseconds." Our techniques, which are based on symbolic computations with vectors and matrices of rational functions, begin with a description of a particular property to be analyzed, and examine a system of PIOAs one component at a time. As each component is treated, a minimization step is performed to eliminate irrelevant information. Thus, our technique amounts to a kind of "partial evaluation" method that retains at each stage only information relevant to the particular property of interest, rather than first calculating a large description of the complete system and then extracting a particular property from this description. We expect that, for the analysis of large systems, our compositional techniques will be a substantial improvement over non-compositional algorithms that require the construction of the full global state space. This expectation seems to be supported by the examples we have analyzed to date.

Our methods for compositional analysis of systems of PIOAs are based on a certain kind of abstract representation of a PIOA, which we call a "behavior" because it represents information about a PIOA that is relevant to its externally observable behavior, while abstracting from specific internal details such as the precise number of states. These PIOA behaviors retain sufficient information, though, that it is possible to extract the performance parameters we are interested in from the behavior without referring to the full PIOA. In addition, behaviors are a compositional representation of PIOAs, in the sense that the behavior of the composition of a compatible collection of PIOAs is completely determined by the behaviors of the components.

Implementation and Initial Benchmarking of our Compositional Methods

We have constructed a prototype implementation of the compositional analysis method described in the previous section, and have done some initial validation and performance tuning on it. The implementation currently consists of about 8200 lines of code in the very high-level functional programming language Standard ML. To date, we have been testing the system on example problems for which we know the correct answers, and for which there are parameters that can be adjusted to obtain a variety of problem sizes. We are now reaching a point where we have some degree of confidence in the implementation, and plan soon to try it out on some realistic examples of practical interest. One example we have been using to test the system is a model of an *n*-player jai-alai match in which *p* points are required to win. The current version of the system will correctly compute the exact rational probability (1/2) that the first player will win in a 2-player, 2-point match, assuming that the two players have equal abilities. The match is modeled by five component automata having 5, 7, 7, 7, and 7, states, respectively, so that a global analysis of this model would thus need to treat 12,005 global states. Our methods achieve substantial reductions with each minimization step, so that the largest number of "dimensions" (the relevant measure of problem size for our technique, roughly comparable to number of states) treated during the analysis is 1351. The analysis takes 2 minutes on a 333MHz Sparc Ultra. Perhaps more interesting from an engineering point of view is that our code can also be used to determine a symbolic expression for the probability the first player will win the match, as a function of the probability *p* that this player will win each point, and it takes only slightly longer to do so. Our code produces the following polynomial expression for this probability:

```
2p**3 - 3p**2 + 2p
```

in 2.5 minutes of compute time. We have also run our analysis algorithm to completion on a model of a 3-player, 2-point jai-alai match. This model has 7 component automata: A1, A2, ..., A7 having, respectively, 5, 7, 8, 7, 8, 7, 8 states, for a total of 878,080 global states. Our algorithm takes 249 minutes to compute the exact rational probability (3/8) that the first player will win the match. The largest representation encountered in this calculation has dimension 13,000. Data from this run is interesting, because it shows the kind of reductions that are achieved at each stage by the minimization algorithm. The figure below shows, as each component automaton is applied in succession, the dimension of the new representation that results from applying the behavior of that automaton to the representation produced at the previous stage, and also the dimension of the representation that results after the minimization algorithm is applied. At each stage, the number of global states of the composition of the components up to and including that stage is shown for comparison purposes.

Component	States	Start dim.	Min. dim	Global states
A1	5	15	6	5
A2	7	42	26	35
A3	8	208	57	280
A4	7	399	281	1,960
A5	8	2248	407	15,680
A6	7	2849	1625	109,760
A7	8	13,000	59	878,080

Performance of Minimization Algorithm

Though the calculation for this example takes a long time, it is important to note that it is entirely CPU-bound and not memory-bound. This is encouraging, as we feel there are still additional possibilities for substantial reductions in the CPU time required by the algorithm. In contrast, memory consumption, which typically tends to be the limiting factor in global, non-compositional analysis methods, can be much harder to reduce.

Verification of the Northrop Grumman E-2C Reliable Broadcast Protocol

The E-2C Hawkeye is an airborne early warning/command and control aircraft developed by the Northrop Grumman Corporation. The Hawkeye can monitor 6 million cubic miles of air space and more than 150,000 square miles of ocean surface for the presence of aircraft, missiles, ships, and fixed targets.

The Hawkeye's Cooperative Engagement Center (CEC) mission computer enables the Hawkeye to serve as the Atlantic fleet's information hub, fusing information from sources such as satellite and shipborne radar, and then distributing that information to those who need it. The CEC is configured as a client/server architecture, with the Mission Computer (MC) as the server and (currently) three Tactical Workstations (TWSs) as the clients.

The communications protocol underlying the CEC, the Display LAN protocol, runs on single-segment local area network, and sits above TCP/IP and UDP/IP. All command and acknowledgement packets are transmitted by TCP/IP, which is reliable but has high overhead. Data packets are transmitted by UDP/IP, which is less reliable but has low overhead. The core of the protocol is its acknowledgement and retransmission scheme.

Since December 1996, we have met several times with Stephanie White and the designers of the Display LAN protocol. It was agreed that we would attempt to formally model the protocol and verify certain key properties using a computer-aided verification tool. To this end, we wrote an abstract specification of the protocol in Promela, the input language of the Bell Labs SPIN verification tool.

Our analysis of the CEC protocol revealed a race condition that resulted in the disconnect of a TWS in the presence of one lost message. Although of interest to the protocol designers, it was subsequently determined that the disconnect had no practical impact on the protocol's operation.

PERSONNEL SUPPORTED

- Scott Smolka
- Eugene Stark
- Giri Pemmasani (Ph.D. student)

PUBLICATIONS

1. E.W. Stark and S.A. Smolka, "Compositional Analysis of Expected Delays in Networks of Probabilistic I/O Automata," *Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science* (LICS '98), Indianapolis, IN, IEEE Computer Society Press (June 1998).

INTERACTIONS/TRANSITIONS

Participation/Presentations At Meetings, Conferences, Seminars, Etc.

1. Scott Smolka was the organizer of the IFIP Working Conference on Programming Concepts and Methods (PROCOMET '98), held on Shelter Island, NY, June 1998.
2. Conference paper presented by Eugene Stark at *LICS '98*, Indianapolis, IN, July 1998. Title of paper: "Compositional Analysis of Expected Delays in Networks of Probabilistic I/O Automata," co-authored with Scott Smolka.
3. Presentation: "Compositional Analysis of Expected Delay in Networks of Automata," made by Eugene Stark at AFOSR PI meeting, Rome, NY, September 1997.

Consultative And Advisory Functions To Other Laboratories And Agencies

None.

Transitions

None.

NEW DISCOVERIES, INVENTIONS, OR PATENT DISCLOSURES

None.

HONORS/AWARDS

None.

Scott A. Smolka, Eugene W. Stark
Last modified: Mon Nov 30 09:38:32 EST 1998